SIAM J. SCI. COMPUT.
Vol. 41, No. 6, pp. A3576–A3595

HIGH-INDEX OPTIMIZATION-BASED SHRINKING DIMER METHOD FOR FINDING HIGH-INDEX SADDLE POINTS*

JIANYUAN YIN†, LEI ZHANG‡, AND PINGWEN ZHANG§

Abstract. We present a high-index optimization-based shrinking dimer (HiOSD) method to compute index-k saddle points as a generalization of the optimization-based shrinking dimer method for index-1 saddle points [L. Zhang, Q. Du, and Z. Zheng, *SIAM J. Sci. Comput.*, 38 (2016), pp. A528–A544]. We first formulate a minimax problem for an index-k saddle point that is a local maximum on a k-dimensional manifold and a local minimum on its orthogonal complement. The k-dimensional maximal subspace is spanned by the k eigenvectors corresponding to the smallest k eigenvalues of the Hessian, which can be constructed by the simultaneous Rayleigh-quotient minimization technique or the locally optimal block preconditioned conjugate gradient method. Under the minimax framework, we implement the Barzilai–Borwein gradient method to speed up the convergence. We demonstrate the efficiency of the HiOSD method for computing high-index saddle points by applying finite-dimensional examples and semilinear elliptic problems.

Key words. rare event, saddle point, Morse index, dimer method, minimax

AMS subject classifications. 37M05, 49K35, 37N30, 34K28, 65P99

DOI. 10.1137/19M1253356

1. Introduction.

The problems of finding saddle points on a potential energy surface (PES) have attracted much attention in different scientific communities during the past decades. A large spectrum of examples include finding critical nuclei and transition pathways in phase transformations [6, 35, 38, 41, 42], computing transition rates in chemical reactions [1, 25], and biology [37, 39, 44]. For instance, the transition state is characterized as an index-1 saddle point, that is, a critical point where the Hessian has one and only one negative eigenvalue. Because of the unstable nature of saddle points, methods for computing saddle points prove to be more challenging than those for a minimization problem.

A large number of numerical algorithms for computing index-1 saddle points have been proposed and applied to many practical problems. Generally speaking, there are two distinct approaches: path-finding methods and surface-walking methods. The former class requires initial and final states as two ends of the pathway for computing the minimum energy path (MEP) on the PES, and the index-1 saddle point corresponds to the configuration with the highest energy on the MEP. The representative methods include the nudged elastic band method [23, 24, 36] and the string method [8, 10, 11, 34]. The latter class starts from a single state on the PES and requires the first or the second derivatives of the potential energy to locate index-1 saddle points without a priori knowledge of the final state. Some samplers of the

*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 29, 2019; accepted for publication (in revised form) September 24, 2019; published electronically November 12, 2019.

https://doi.org/10.1137/19M1253356

Funding: This work was supported by the National Natural Science Foundation of China through grants 11622102, 11421110001, 11861130351, 11421101, and 21790340. The first author is partially supported by the Elite Program of Computational and Applied Mathematics for PhD candidates in Peking University.

†School of Mathematical Sciences, Peking University, Beijing 100871, China (yinjy@pku.edu.cn).

‡Beijing International Center for Mathematical Research, Center for Quantitative Biology, Peking University, Beijing 100871, China (zhangl@math.pku.edu.cn).

§LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China (pzhang@pku.edu.cn).

surface-walking methods are the gentlest ascent dynamics [13, 15, 19], the dimer-type methods [17, 22, 23, 40, 43], the minimax method [27], the activation-relaxation technique [4, 29], the trajectory following algorithm [18], the step and slide method [31], the eigenvector-following method [5], the biased gradient squared descent method [9], etc. We refer to [12, 21, 45] as some excellent reviews.

While most existing algorithms are designed to find index-1 saddle points, the problems of computing high-index saddle points have received little attention despite its important applications. For instance, index-2 saddle points are particularly interesting in chemical systems for providing valuable information on the trajectories of chemical reactions [20]. The excited states in quantum mechanics that have higher energy than the ground states can also be characterized as high-index saddle point configurations because they could return to lower-energy excited states or ground states after excitation [14]. In terms of numerical algorithms, the path-finding methods can only find index-1 saddle points because the only unstable (maximum) direction is the direction along the MEP, and the other orthometric directions lead to the minimum. On the other hand, the surface-walking methods allow several approaches to find high-index saddle points. For instance, the minimax method [27] applied the local minimax theorem to compute index-k saddle points but required a priori knowledge of index-$(k-1)$ saddle points. The biased gradient squared descent method [9] can also find high-index saddle points by transforming all critical points of the original potential energy into the minima of a gradient squared landscape, but the index of saddle points cannot be determined before calculation. The gentlest ascent dynamics (GAD) was developed by E and Zhou to search index-1 saddle points as a dynamical system of the gentlest ascent method [7], and such formulation can be extended to compute index-2 saddle points [13]. Quapp and Bofill developed a generalized GAD algorithm to locate high-index saddle points on the PES. It regards the saddle-searching problem as a dynamical system and requires the explicit calculation of the Hessian matrix [33].

Recently, Zhang, Du, and Zheng proposed the optimization-based shrinking dimer (OSD) method to efficiently compute index-1 saddle points [43], which can be regarded as a generalized formulation of the shrinking dimer dynamics [40]. The dimer rotation and translation steps in the dimer method [22] are transformed into corresponding optimization problems. Under the optimization framework, the OSD method shows a great advantage of applying optimization algorithms to accelerate the convergence. By adopting the similar spirit of the OSD, we present a high-index optimization-based shrinking dimer (HiOSD) method to search index-k saddle points. We first formulate the minimax problem for an index-k saddle point and then construct the maximal subspace by minimizing Rayleigh quotients simultaneously. Thus the HiOSD dynamical system is presented for finding an index-k saddle point, and the stability analysis is performed to show the linearly stable steady states of the HiOSD system are exactly index-k saddle points. To improve the numerical approximations of eigenvectors, we provide an alternative way by adopting the locally optimal block preconditioned conjugate gradient (LOBPCG) method to construct the maximal subspace. Furthermore, we apply the Barzilai–Borwein (BB) gradient method [3] to determine the step sizes in order to accelerate the convergence. Numerical examples, including finite-dimensional problems and semilinear elliptic problems, are presented to illustrate the efficiency of the HiOSD method for searching index-k saddle points.

2. HiOSD method. Given a twice Fréchet differentiable energy functional $E(\boldsymbol{x})$ defined on a real Hilbert space $\mathcal{H}$ with an inner product $\langle\cdot,\cdot\rangle$, we let $\boldsymbol{F}(\boldsymbol{x}) = -\nabla E(\boldsymbol{x})$ denote its natural force and $\mathbb{G}(\boldsymbol{x}) = \nabla^2 E(\boldsymbol{x})$ denote its Hessian. By the Riesz

representation theorem, we regard $\boldsymbol{F}(\boldsymbol{x})$ as an element of $\mathcal{H}$. $\hat{\boldsymbol{x}} \in \mathcal{H}$ is called a *critical point* of $E(\boldsymbol{x})$ if $\|\boldsymbol{F}(\hat{\boldsymbol{x}})\| = 0$. A critical point of $E(\boldsymbol{x})$ that is not a local extremum is called a *saddle point* of $E(\boldsymbol{x})$. A critical point $\hat{\boldsymbol{x}}$ is called *nondegenerate* if $\mathbb{G}(\hat{\boldsymbol{x}})$ has a bounded inverse. According to the Morse theory [30], the *index* (Morse index) of a nondegenerate critical point $\hat{\boldsymbol{x}}$ is the maximal dimension of a subspace $\mathcal{K}$ on which the operator $\mathbb{G}(\hat{\boldsymbol{x}})$ is negative definite. Our aim is to find an index-k saddle point, or for short, a *k-saddle*, on the PES. For simplicity, we assume the dimension of $\mathcal{H}$ is d and write the inner product $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ as $\boldsymbol{x}^\top \boldsymbol{y}$.

2.1. Minimax optimization for a k-saddle. For a nondegenerate k-saddle $\hat{\boldsymbol{x}}$, the Hessian $\mathbb{G}(\hat{\boldsymbol{x}})$ has exactly k negative eigenvalues $\hat{\lambda}_1 \leqslant \cdots \leqslant \hat{\lambda}_k$ with corresponding unit eigenvectors $\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_k$ satisfying $\langle \hat{\boldsymbol{v}}_j, \hat{\boldsymbol{v}}_i\rangle = \delta_{ij}$, $1 \leqslant i, j \leqslant k$. By setting a k-dimensional subspace $\hat{\mathcal{V}} = \mathrm{span}\{\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_k\}$, $\hat{\boldsymbol{x}}$ is a local maximum on a k-dimensional linear manifold $\hat{\boldsymbol{x}} + \hat{\mathcal{V}}$ and a local minimum on $\hat{\boldsymbol{x}} + \hat{\mathcal{V}}^\perp$, where $\hat{\mathcal{V}}^\perp$ is the orthogonal complement space of $\hat{\mathcal{V}}$. By decomposing the Hilbert space $\mathcal{H}$ as $\hat{\mathcal{V}} \oplus \hat{\mathcal{V}}^\perp$, the energy functional E can be defined with two inputs, respectively, in $\hat{\mathcal{V}}$ and $\hat{\mathcal{V}}^\perp$ naturally as $E(\boldsymbol{v}, \boldsymbol{w}) = E(\boldsymbol{v} + \boldsymbol{w})$, where $\boldsymbol{v} \in \hat{\mathcal{V}}$ and $\boldsymbol{w} \in \hat{\mathcal{V}}^\perp$. Therefore, the k-saddle $\hat{\boldsymbol{x}} = (\hat{\boldsymbol{x}}_{\hat{\mathcal{V}}}, \hat{\boldsymbol{x}}_{\hat{\mathcal{V}}^\perp})$ is a solution to the following minimax optimization problem:

$$\min_{\boldsymbol{w} \in \hat{\mathcal{V}}^\perp} \max_{\boldsymbol{v} \in \hat{\mathcal{V}}} E(\boldsymbol{v}, \boldsymbol{w}), \tag{1}$$

where $\hat{\boldsymbol{x}}_{\hat{\mathcal{V}}}$ is a local maximum of $E(\cdot, \hat{\boldsymbol{x}}_{\hat{\mathcal{V}}^\perp})$ on $\hat{\mathcal{V}}$, and $\hat{\boldsymbol{x}}_{\hat{\mathcal{V}}^\perp}$ is a local minimum of $E(\hat{\boldsymbol{x}}_{\hat{\mathcal{V}}}, \cdot)$ on $\hat{\mathcal{V}}^\perp$. Such a splitting structure exists around the nondegenerate k-saddle $\hat{\boldsymbol{x}}$ according to the Morse theory [27, 30]. It should be noticed that the subspace $\hat{\mathcal{V}}$ in (1) is unknown, which is an intrinsic difference from the classical minimax problems in optimization.

We first assume that $\boldsymbol{x}$ is sufficiently close to $\hat{\boldsymbol{x}}$ so that $\mathbb{G}(\boldsymbol{x})$ has also exactly k negative eigenvalues. Let $\{(\lambda_i, \boldsymbol{v}_i)_{i=1}^k : \lambda_1 \leqslant \cdots \leqslant \lambda_k < 0, \langle \boldsymbol{v}_j, \boldsymbol{v}_i\rangle = \delta_{ij}\}$ denote the eigenpairs of $\mathbb{G}(\boldsymbol{x})$, and we use a subspace $\mathcal{V} = \mathrm{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ to approximate $\hat{\mathcal{V}}$ in (1). We define $\mathcal{P}_{\mathcal{V}}$ as the orthogonal projection operator on the finite-dimensional subspace $\mathcal{V} \subseteq \mathcal{H}$. To find a solution of the minimax problem (1), the dynamics of $\boldsymbol{x}$ is supposed to satisfy that $\mathcal{P}_{\mathcal{V}}\dot{\boldsymbol{x}}$ is an ascent direction on the subspace $\mathcal{V}$ and that $\dot{\boldsymbol{x}} - \mathcal{P}_{\mathcal{V}}\dot{\boldsymbol{x}}$ is a descent direction on the subspace $\mathcal{V}^\perp$. Therefore, we take $-\mathcal{P}_{\mathcal{V}}\boldsymbol{F}(\boldsymbol{x})$ as the direction of $\mathcal{P}_{\mathcal{V}}\dot{\boldsymbol{x}}$ and $\boldsymbol{F}(\boldsymbol{x}) - \mathcal{P}_{\mathcal{V}}\boldsymbol{F}(\boldsymbol{x})$ as the direction of $\dot{\boldsymbol{x}} - \mathcal{P}_{\mathcal{V}}\dot{\boldsymbol{x}}$, obtaining a gradient dynamics as

$$\dot{\boldsymbol{x}} = \beta_{\mathcal{V}}\left(-\mathcal{P}_{\mathcal{V}}\boldsymbol{F}(\boldsymbol{x})\right) + \beta_{\mathcal{V}^\perp}\left(\boldsymbol{F}(\boldsymbol{x}) - \mathcal{P}_{\mathcal{V}}\boldsymbol{F}(\boldsymbol{x})\right), \tag{2}$$

where $\beta_{\mathcal{V}}$ and $\beta_{\mathcal{V}^\perp}$ are positive relaxation constants. Alternatively, we may take a modified direction by choosing $\beta_{\mathcal{V}} = \beta_{\mathcal{V}^\perp} = \beta$ to get a modified gradient dynamics for a k-saddle as

$$\beta^{-1}\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}) - 2\mathcal{P}_{\mathcal{V}}\boldsymbol{F}(\boldsymbol{x}). \tag{3}$$

2.2. Construction of the subspace $\mathcal{V}$. The subspace $\mathcal{V}$ in (3) is constructed by approximating the k eigenvectors simultaneously, and the approximate eigenvectors should also satisfy the orthonormal condition

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle = \delta_{ij}, \quad i, j = 1, 2, \ldots, k. \tag{4}$$

The eigenvector $\boldsymbol{v}_1$ corresponding to the smallest eigenvalue λ_1 can be obtained by optimizing the Rayleigh quotient $\langle \boldsymbol{v}_1, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_1\rangle / \langle \boldsymbol{v}_1, \boldsymbol{v}_1\rangle$, or simply $\langle \boldsymbol{v}_1, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_1\rangle$ with

a constraint $\langle \boldsymbol{v}_1, \boldsymbol{v}_1 \rangle = 1$. In a similar manner, computing the ith eigenvector $\boldsymbol{v}_i$ can be transformed into a constrained optimization problem with the knowledge of $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{i-1}$,

$$\min_{\boldsymbol{v}_i} \quad \langle \boldsymbol{v}_i, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i \rangle \qquad \text{s.t.} \quad \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = \delta_{ij}, \quad j = 1, 2, \ldots, i. \tag{5}$$

Then we minimize the k Rayleigh quotients (5) simultaneously [28] by using gradient-type dynamics of $\boldsymbol{v}_i$ as follows:

$$\gamma^{-1}\dot{\boldsymbol{v}}_i = -\mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i + \langle \boldsymbol{v}_i, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i \rangle \boldsymbol{v}_i + \sum_{j=1}^{i-1} 2 \langle \boldsymbol{v}_j, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i \rangle \boldsymbol{v}_j, \quad i = 1, 2, \ldots, k, \tag{6}$$

where $\gamma > 0$ is a relaxation parameter. To derive (6), the Lagrangian function of (5) is

$$\mathcal{L}_i(\boldsymbol{v}_i; \xi_1, \ldots, \xi_{i-1}, \xi_i) = \langle \boldsymbol{v}_i, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i \rangle - \xi_i (\langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle - 1) - \sum_{j=1}^{i-1} \xi_j \langle \boldsymbol{v}_j, \boldsymbol{v}_i \rangle, \tag{7}$$

where $\xi_1, \ldots, \xi_i$ are Lagrangian multipliers, and the gradient of (7) is

$$\frac{\partial}{\partial \boldsymbol{v}_i}\mathcal{L}_i(\boldsymbol{v}_i; \xi_1, \ldots, \xi_{i-1}, \xi_i) = 2\mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i - 2\xi_i \boldsymbol{v}_i - \sum_{j=1}^{i-1} \xi_j \boldsymbol{v}_j. \tag{8}$$

First, the dynamics for $\boldsymbol{v}_1$ is

$$\dot{\boldsymbol{v}}_1 = -\frac{\gamma}{2}\frac{\partial}{\partial \boldsymbol{v}_1}\mathcal{L}_1(\boldsymbol{v}_1; \xi_1) = -\gamma (\mathbb{G}(\boldsymbol{x})\boldsymbol{v}_1 - \xi_1 \boldsymbol{v}_1). \tag{9}$$

Since $\langle \boldsymbol{v}_1, \dot{\boldsymbol{v}}_1 \rangle = 0$ should be satisfied from $\langle \boldsymbol{v}_1, \boldsymbol{v}_1 \rangle = 1$, we have $\xi_1 = \langle \boldsymbol{v}_1, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_1 \rangle$, indicating the $i = 1$ case of (6). With $i \leqslant m-1$ cases of (6), we set the dynamics of $\boldsymbol{v}_m$ as

$$\dot{\boldsymbol{v}}_m = -\frac{\gamma}{2}\frac{\partial}{\partial \boldsymbol{v}_m}\mathcal{L}_m(\boldsymbol{v}_m; \xi_1, \ldots, \xi_{m-1}, \xi_m) = -\gamma \left(\mathbb{G}(\boldsymbol{x})\boldsymbol{v}_m - \xi_m \boldsymbol{v}_m - \frac{1}{2}\sum_{j=1}^{m-1} \xi_j \boldsymbol{v}_j \right). \tag{10}$$

$\xi_m = \langle \boldsymbol{v}_m, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_m \rangle$ is derived from $\langle \boldsymbol{v}_m, \dot{\boldsymbol{v}}_m \rangle = 0$ similarly. From the condition $\langle \boldsymbol{v}_j, \boldsymbol{v}_m \rangle = 0$ for $j = 1, \ldots, m-1$, the dynamics (10) should satisfy $\langle \dot{\boldsymbol{v}}_j, \boldsymbol{v}_m \rangle + \langle \boldsymbol{v}_j, \dot{\boldsymbol{v}}_m \rangle = 0$, indicating $\xi_j = 4\langle \boldsymbol{v}_j, \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_m \rangle$. As a consequence of introduction, we obtain the dynamics (6).

Under the orthonormal condition (4), the projection operator $\mathcal{P}_{\mathcal{V}}$ has a simple form of $\sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^\top$, and we obtain a dynamical system for a k-saddle:

$$\begin{cases} \beta^{-1}\dot{\boldsymbol{x}} = \left(\mathbb{I} - \sum_{i=1}^{k} 2\boldsymbol{v}_i \boldsymbol{v}_i^\top \right) \boldsymbol{F}(\boldsymbol{x}), \\ \gamma^{-1}\dot{\boldsymbol{v}}_i = -\left(\mathbb{I} - \boldsymbol{v}_i \boldsymbol{v}_i^\top - \sum_{j=1}^{i-1} 2\boldsymbol{v}_j \boldsymbol{v}_j^\top \right) \mathbb{G}(\boldsymbol{x})\boldsymbol{v}_i, \quad i = 1, 2, \ldots, k, \end{cases} \tag{11}$$

where $\mathbb{I}$ is the identity operator and $\beta, \gamma > 0$ are relaxation parameters.

Because Hessians are often expensive to compute and store, we use first derivatives to approximate the Hessians in (11) by k shrinking dimers centered at $\boldsymbol{x}$, which is essentially central difference schemes for directional derivatives. The ith dimer $(i = 1, \ldots, k)$ has a direction of $\boldsymbol{v}_i$ and length $2l$ and $\mathbb{G}(\boldsymbol{x})\boldsymbol{v}$ is approximated by

$$\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}, l) = -\frac{\boldsymbol{F}(\boldsymbol{x} + l\boldsymbol{v}) - \boldsymbol{F}(\boldsymbol{x} - l\boldsymbol{v})}{2l}. \tag{12}$$

By adopting the idea of the shrinking dimer dynamics [40], we shrink the dimers by using a simple dynamics $\dot{l} = -l$, which corresponds to an exponential decay of the dimer length. In numerical algorithms, we set a minimum dimer length to avoid numerical errors due to round-off.

Now we obtain a complete HiOSD dynamical system for a k-saddle,

$$\begin{cases} \beta^{-1}\dot{\boldsymbol{x}} = \left(\mathbb{I} - \sum_{i=1}^{k} 2\boldsymbol{v}_i\boldsymbol{v}_i^\top\right)\boldsymbol{F}(\boldsymbol{x}), \\ \gamma^{-1}\dot{\boldsymbol{v}}_i = -\left(\mathbb{I} - \boldsymbol{v}_i\boldsymbol{v}_i^\top - \sum_{j=1}^{i-1} 2\boldsymbol{v}_j\boldsymbol{v}_j^\top\right)\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}_i, l), \quad i = 1, 2, \ldots, k, \\ \dot{l} = -l \end{cases} \tag{13}$$

with the orthonormal condition (4) satisfied initially.

Next we show that linearly stable steady states of (13) are exactly k-saddles.

THEOREM 1. *Assume that $E(\boldsymbol{x})$ is a $\mathcal{C}^3$ functional, $\boldsymbol{x}^* \in \mathcal{H}, \{\boldsymbol{v}_i^*\}_{i=1}^k \subset \mathcal{H}$ satisfies $\|\boldsymbol{v}_i^*\| = 1$, the Hessian $\mathbb{G}^* = \mathbb{G}(\boldsymbol{x}^*)$ is nondegenerate, whose eigenvalues are $\lambda_1^* < \cdots < \lambda_k^* \leqslant \lambda_{k+1}^* \leqslant \cdots \leqslant \lambda_d^*$, and $\beta, \gamma > 0$; then $(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*)$ is a linear stable steady state of* (13) *if and only if $\boldsymbol{x}^*$ is a k-saddle of $E(\boldsymbol{x})$, $\boldsymbol{v}_i^*$ is an eigenvector of $\mathbb{G}^*$ corresponding to the eigenvalue λ_i^*, and l^* is* 0.

Proof. Taking the derivative of (12) with respect to l, we have

$$\frac{\partial}{\partial l}\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}, l) = \frac{\mathbb{G}(\boldsymbol{x} + l\boldsymbol{v}) + \mathbb{G}(\boldsymbol{x} - l\boldsymbol{v})}{2l}\boldsymbol{v} + \frac{\boldsymbol{F}(\boldsymbol{x} + l\boldsymbol{v}) - \boldsymbol{F}(\boldsymbol{x} - l\boldsymbol{v})}{2l^2}. \tag{14}$$

As l decreases to zero, by the smoothness of $E(\boldsymbol{x})$ we have

$$\frac{\partial}{\partial l}\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}, l) = \frac{\mathbb{G}(\boldsymbol{x})\boldsymbol{v} + o(l)}{l} + \frac{-2l\mathbb{G}(\boldsymbol{x})\boldsymbol{v} + o(l^2)}{2l^2} = o(1) \quad (\text{as } l \to 0). \tag{15}$$

We consider the Jacobian operator of the dynamics (13),

$$\mathbb{J} = \frac{\partial(\dot{\boldsymbol{x}}, \dot{\boldsymbol{v}}_1, \dot{\boldsymbol{v}}_2, \ldots, \dot{\boldsymbol{v}}_k, \dot{l})}{\partial(\boldsymbol{x}, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k, l)} = \begin{pmatrix} \mathbb{J}_{\boldsymbol{x}} & \mathbb{J}_{\boldsymbol{x}1} & \mathbb{J}_{\boldsymbol{x}2} & \cdots & \mathbb{J}_{\boldsymbol{x}k} & \mathbb{O} \\ \mathbb{J}_{1\boldsymbol{x}} & \mathbb{J}_1 & \mathbb{O} & \cdots & \mathbb{O} & \mathbb{J}_{1l} \\ \mathbb{J}_{2\boldsymbol{x}} & \mathbb{J}_{21} & \mathbb{J}_2 & \cdots & \mathbb{O} & \mathbb{J}_{2l} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \mathbb{J}_{k\boldsymbol{x}} & \mathbb{J}_{k1} & \mathbb{J}_{k2} & \cdots & \mathbb{J}_k & \mathbb{J}_{kl} \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \cdots & \mathbb{O} & -1 \end{pmatrix}, \tag{16}$$

whose blocks have following expressions:

$$\mathbb{J}_{\boldsymbol{x}} = \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{x}} = -\beta \left(\mathbb{I} - \sum_{i=1}^{k} 2\boldsymbol{v}_i \boldsymbol{v}_i^\top \right) \mathbb{G}(\boldsymbol{x}), \tag{17}$$

$$\mathbb{J}_{\boldsymbol{x}i} = \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{v}_i} = -2\beta \left(\boldsymbol{v}_i^\top \boldsymbol{F}(\boldsymbol{x}) \mathbb{I} + \boldsymbol{v}_i \boldsymbol{F}(\boldsymbol{x})^\top \right), \tag{18}$$

$$\begin{aligned} \mathbb{J}_i = \frac{\partial \dot{\boldsymbol{v}}_i}{\partial \boldsymbol{v}_i} = &-\gamma \left(\mathbb{I} - \boldsymbol{v}_i \boldsymbol{v}_i^\top - \sum_{j=1}^{i-1} 2\boldsymbol{v}_j \boldsymbol{v}_j^\top \right) \frac{\mathbb{G}(\boldsymbol{x} + l\boldsymbol{v}) + \mathbb{G}(\boldsymbol{x} - l\boldsymbol{v})}{2} \\ &+ \gamma \left(\boldsymbol{v}_i^\top \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}_i, l) \mathbb{I} + \boldsymbol{v}_i \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}_i, l)^\top \right), \end{aligned} \tag{19}$$

$$\mathbb{J}_{il} = \frac{\partial \dot{\boldsymbol{v}}_i}{\partial l} = -\gamma \left(\mathbb{I} - \boldsymbol{v}_i \boldsymbol{v}_i^\top - \sum_{j=1}^{i-1} 2\boldsymbol{v}_j \boldsymbol{v}_j^\top \right) \partial_l \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{v}_i, l). \tag{20}$$

Furthermore, $\mathbb{J}_{\boldsymbol{x}i}$ is null if $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$, and $\mathbb{J}_{il}$ is null if $l = 0$.

"$\Leftarrow$": Supposing that $\boldsymbol{x}^*$ is a k-saddle of $E(\boldsymbol{x})$, $(\lambda_i^*, \boldsymbol{v}_i^*)$ is an eigenpair of $\mathbb{G}^*$, and l^* is zero, we have $\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{0}$ and $\boldsymbol{H}(\boldsymbol{x}^*, \boldsymbol{v}_i^*, l^*) = \mathbb{G}^* \boldsymbol{v}_i^* = \lambda_i^* \boldsymbol{v}_i^*$, so $(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*)$ is a steady state of (13).

In order to show the linear stability, we consider the spectrum of $\mathbb{J}^* = \mathbb{J}(\boldsymbol{x}^*)$. From $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ and $l = 0$, $\mathbb{J}_{\boldsymbol{x}i}$ and $\mathbb{J}_{il}$ are null, so $\mathbb{J}^*$ is block lower triangular. The first diagonal block,

$$\mathbb{J}_{\boldsymbol{x}}(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*) = \beta \left(-\mathbb{G}^* + \sum_{i=1}^{k} 2\lambda_i^* \boldsymbol{v}_i^* \boldsymbol{v}_i^{*\top} \right), \tag{21}$$

has eigenvalues of $\{\beta\lambda_1^*, \ldots, \beta\lambda_k^*, -\beta\lambda_{k+1}^*, \ldots, -\beta\lambda_d^*\}$, and the diagonal block,

$$\mathbb{J}_i(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*) = \gamma \left(-\mathbb{G}^* + \sum_{j=1}^{i} 2\lambda_j^* \boldsymbol{v}_j^* \boldsymbol{v}_j^{*\top} + \lambda_i^* \mathbb{I} \right), \tag{22}$$

has eigenvalues of $\{\gamma(\lambda_i^* + \lambda_1^*), \ldots, \gamma(\lambda_i^* + \lambda_i^*), \gamma(\lambda_i^* - \lambda_{i+1}^*), \ldots, \gamma(\lambda_i^* - \lambda_d^*)\}$. Because all eigenvalues of $\mathbb{J}^*$ are negative, $(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*)$ is linear stable.

"$\Rightarrow$": Supposing that $(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*)$ is a linear stable steady state, we have $l^* = 0$ straightforward, and therefore,

$$\left(\mathbb{I} - \boldsymbol{v}_i^* \boldsymbol{v}_i^{*\top} - \sum_{j=1}^{i-1} 2\boldsymbol{v}_j^* \boldsymbol{v}_j^{*\top} \right) \mathbb{G}^* \boldsymbol{v}_i^* = \boldsymbol{0}, \quad i = 1, 2, \ldots, k. \tag{23}$$

Defining $\mu_i^* = \langle \boldsymbol{v}_i^*, \mathbb{G}^* \boldsymbol{v}_i^* \rangle$, we now show that for $i = 1, \ldots, k$,

$$\mathbb{G}^* \boldsymbol{v}_i^* = \mu_i^* \boldsymbol{v}_i^*; \quad \left\langle \boldsymbol{v}_j^*, \boldsymbol{v}_i^* \right\rangle = \delta_{ij}, \quad j = 1, 2, \ldots, i-1, \tag{24}$$

by introduction. Taking $i = 1$ in (23), we have $\mathbb{G}^* \boldsymbol{v}_1^* = \mu_1^* \boldsymbol{v}_1^*$ straightforward. Assuming that (24) holds for $i = 1, \ldots, m-1$, by taking $i = m$ in (23), we have

$$\left(\mathbb{G}^* - \sum_{j=1}^{m-1} 2\mu_j^* \boldsymbol{v}_j^* \boldsymbol{v}_j^{*\top} \right) \boldsymbol{v}_m^* = \mu_m^* \boldsymbol{v}_m. \tag{25}$$

Since $\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_{m-1}^*$ are eigenvectors of $\mathbb{G}^*$ according to the inductive assumption, $\mathbb{G}^* - \sum_{j=1}^{m-1} 2\mu_j^* \boldsymbol{v}_j^* \boldsymbol{v}_j^{*\top}$ and $\mathbb{G}^*$ share the same eigenvectors, so $\boldsymbol{v}_m^*$ is an eigenvector of $\mathbb{G}^*$ as well, and the eigenvalue is therefore μ_m^* by definition. From $\mathbb{G}^* \boldsymbol{v}_m^* = \mu_m^* \boldsymbol{v}_m^*$ and (25), we obtain $\sum_{j=1}^{m-1} \mu_j^* \langle \boldsymbol{v}_j^*, \boldsymbol{v}_m^* \rangle \boldsymbol{v}_j^* = 0$. Because $\{\boldsymbol{v}_j^*\}_{j=1}^{m-1}$ are orthogonal to each other, and the eigenvalue $\mu_j^* \neq 0$, we have $\langle \boldsymbol{v}_j^*, \boldsymbol{v}_m^* \rangle = 0$ for $j = 1, \ldots, m-1$. As a consequence of introduction, we prove the property (24). Furthermore, we have $\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{0}$ from $(\mathbb{I} - \sum_{i=1}^{k} 2\boldsymbol{v}_i^* \boldsymbol{v}_i^{*\top})^2 = \mathbb{I}$, which indicates that $\boldsymbol{x}^*$ is a critical point of $E(\boldsymbol{x})$.

Similarly, from $l^* = 0$ and $\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{0}$, $\mathbb{J}^*$ is block lower triangular, so all eigenvalues of the diagonal blocks of $\mathbb{J}^*$ are negative. We have proved that $\{(\mu_j^*, \boldsymbol{v}_j^*)\}_{j=1}^{i}$ are eigenpairs of $\mathbb{G}^*$, and then we let $\mu_{k+1}^* \leqslant \cdots \leqslant \mu_d^*$ denote the other eigenvalues of $\mathbb{G}^*$. The eigenvalues of the first diagonal block,

$$\mathbb{J}_{\boldsymbol{x}}(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*) = \beta \left(-\mathbb{G}^* + \sum_{i=1}^{k} 2\mu_i^* \boldsymbol{v}_i^* \boldsymbol{v}_i^{*\top} \right), \tag{26}$$

are $\{\beta\mu_1^*, \ldots, \beta\mu_k^*, -\beta\mu_{k+1}^*, \ldots, -\beta\mu_d^*\}$, so $\{\mu_i^*\}_{i=k+1}^{d}$ are positive, and $\{\mu_i^*\}_{i=1}^{k}$ are negative, which means that $\boldsymbol{x}^*$ is a k-saddle. The eigenvalues of the diagonal block,

$$\mathbb{J}_i(\boldsymbol{x}^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_k^*, l^*) = \gamma \left(-\mathbb{G}^* + \sum_{j=1}^{i} 2\mu_j^* \boldsymbol{v}_j^* \boldsymbol{v}_j^{*\top} + \mu_i^* \mathbb{I} \right), \tag{27}$$

are $\{\gamma(\mu_i^* + \mu_1^*), \ldots, \gamma(\mu_i^* + \mu_i^*), \gamma(\mu_i^* - \mu_{i+1}^*), \ldots, \gamma(\mu_i^* - \mu_d^*)\}$, so μ_i^* is smaller than μ_{i+1}^* for $i = 1, \ldots, k$. Therefore, μ_i^* is exactly λ_i^*, which ends our proof. ∎

We present Algorithm 1 as a discrete scheme for (13).

Algorithm 1. HiOSD method for a k-saddle.

Input: $k \in \mathbb{N}, l^{(0)} > 0, \boldsymbol{x}^{(0)} \in \mathcal{H}, \{\boldsymbol{v}_i^{(0)}\}_{i=1}^{k} \subset \mathcal{H}$ satisfying $\left\langle \boldsymbol{v}_j^{(0)}, \boldsymbol{v}_i^{(0)} \right\rangle = \delta_{ij}$.

Set $n = 0$, compute $\boldsymbol{f}^{(0)} = \boldsymbol{F}(\boldsymbol{x}^{(0)})$;
repeat
  $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \beta^{(n)} \boldsymbol{g}^{(n)}$;
  for $i = 1 : k$ **do**
    $\boldsymbol{v}_i^\star = \boldsymbol{v}_i^{(n)} + \gamma_i^{(n)} \boldsymbol{d}_i^{(n)}$;
    $\boldsymbol{v}_i^\star = \boldsymbol{v}_i^\star - \sum_{j=1}^{i-1} \langle \boldsymbol{v}_j^{(n+1)}, \boldsymbol{v}_i^\star \rangle \boldsymbol{v}_j^{(n+1)}$;
    $\boldsymbol{v}_i^{(n+1)} = \boldsymbol{v}_i^\star / \|\boldsymbol{v}_i^\star\|$;
  end for
  $l^{(n+1)} = \max\left\{ l^{(n)} / (1 + \beta^{(n)}), \varepsilon \right\}$;
  $\boldsymbol{f}^{(n+1)} = \boldsymbol{F}(\boldsymbol{x}^{(n+1)})$;
  $n := n + 1$;
until $\|\boldsymbol{f}^{(n)}\| < \epsilon_F$;

Output: $\boldsymbol{x}^{(n)}, \boldsymbol{v}_1^{(n)}, \ldots, \boldsymbol{v}_k^{(n)}$.

In Algorithm 1, $\boldsymbol{g}^{(n)}$ and $\boldsymbol{d}_i^{(n)}$ are defined as

$$\boldsymbol{g}^{(n)} = \boldsymbol{f}^{(n)} - 2\sum_{i=1}^{k} \left\langle \boldsymbol{v}_i^{(n)}, \boldsymbol{f}^{(n)} \right\rangle \boldsymbol{v}_i^{(n)}, \tag{28}$$

$$\boldsymbol{d}_i^{(n)} = -\boldsymbol{u}_i^{(n)} + \left\langle \boldsymbol{v}_i^{(n)}, \boldsymbol{u}_i^{(n)} \right\rangle \boldsymbol{v}_i^{(n)} + \sum_{j=1}^{i-1} 2 \left\langle \boldsymbol{v}_j^{(n)}, \boldsymbol{u}_i^{(n)} \right\rangle \boldsymbol{v}_j^{(n)}, \tag{29}$$

$$\boldsymbol{u}_i^{(n)} = \boldsymbol{H}\left(\boldsymbol{x}^{(n+1)}, \boldsymbol{v}_i^{(n)}, l^{(n)}\right), \tag{30}$$

and $\beta^{(n)}, \gamma_i^{(n)}$ are the step sizes which will be determined in subsection 2.4. The Gram–Schmidt orthogonalization in step 6 and 7 is used to ensure that $\langle \boldsymbol{v}_j^{(n+1)}, \boldsymbol{v}_i^{(n+1)} \rangle = \delta_{ij}$ holds in discrete schemes. Dimer semilengths are lower-bounded by ε in step 9 for numerical stability, which is set as 10^{-6} in our numerical examples. Furthermore, to reduce the force evaluations, we can compute $\boldsymbol{f}^{(n+1)}$ by

$$\boldsymbol{f}^{(n+1)} = \frac{1}{k}\sum_{i=1}^{k} \frac{\boldsymbol{F}\left(\boldsymbol{x}^{(n+1)} + l^{(n)}\boldsymbol{v}_i^{(n)}\right) + \boldsymbol{F}\left(\boldsymbol{x}^{(n+1)} - l^{(n)}\boldsymbol{v}_i^{(n)}\right)}{2} \tag{31}$$

in step 10, because the forces in the right side of (31) have been calculated in the previous iteration step.

2.3. Construction of subspace $\mathcal{V}$ with LOBPCG. In discrete schemes, there are various efficient algorithms for finding eigenvectors of a large real symmetric matrix, but it is expensive and unnecessary to compute them accurately in each iteration step. To find better approximations of eigenvectors within admissible costs, we introduce the idea of the LOBPCG method [26].

The LOBPCG method is an iterative algorithm to find the smallest (or largest) k eigenvalues (or generalized eigenvalues) of a real symmetric matrix $\mathbb{A}$. Assuming $\{\boldsymbol{v}_i^{(n)}\}_{i=1}^k$ are approximations of k unit eigenvectors at the nth iteration step, we solve an optimization problem,

$$\min \quad \sum_{i=1}^{k} \langle \boldsymbol{v}_i, \mathbb{A}\boldsymbol{v}_i \rangle \quad \text{s.t. } \boldsymbol{v}_i \in \mathcal{U}^{(n)},\ \langle \boldsymbol{v}_j, \boldsymbol{v}_i \rangle = \delta_{ij}, \tag{32}$$

exactly with Rayleigh–Ritz methods to obtained $\{\boldsymbol{v}_i^{(n+1)}\}_{i=1}^k$, where the subspace $\mathcal{U}^{(n)}$ is constructed according to $\{\boldsymbol{v}_i^{(n)}\}_{i=1}^k$. Given a symmetric positive definite preconditioner $\mathbb{T}$, the locally optimal block preconditioned steepest descent (LOBPSD) method chooses a subspace

$$\mathcal{U}_{\mathrm{SD}}^{(n)} = \mathrm{span}\left\{\boldsymbol{v}_i^{(n)}, \mathbb{T}\left(\mathbb{A}\boldsymbol{v}_i^{(n)} - \left\langle \boldsymbol{v}_i^{(n)}, \mathbb{A}\boldsymbol{v}_i^{(n)} \right\rangle \boldsymbol{v}_i^{(n)}\right) : i = 1, \ldots, k\right\}, \tag{33}$$

and the LOBPCG method chooses a larger subspace

$$\mathcal{U}_{\mathrm{CG}}^{(n)} = \mathrm{span}\left\{\boldsymbol{v}_i^{(n-1)}, \boldsymbol{v}_i^{(n)}, \mathbb{T}\left(\mathbb{A}\boldsymbol{v}_i^{(n)} - \left\langle \boldsymbol{v}_i^{(n)}, \mathbb{A}\boldsymbol{v}_i^{(n)} \right\rangle \boldsymbol{v}_i^{(n)}\right) : i = 1, \ldots, k\right\}. \tag{34}$$

Since the matrix $\mathbb{A} = \mathbb{G}(\boldsymbol{x})$ is different at each iteration step and the terms with Hessians are approximated by dimers, we dilate on how the LOBPSD method is

applied in the HiOSD method. We first apply the LOBPSD method in each iteration step and use dimers (12) to approximate $\mathbb{G}(\boldsymbol{x})\boldsymbol{v}$ as well. We define $\boldsymbol{u}_i^{(n)}$ as (30), and

$$\boldsymbol{w}_i^{(n)} = \mathbb{T}\left(\boldsymbol{u}_i^{(n)} - \left\langle \boldsymbol{v}_i^{(n)}, \boldsymbol{u}_i^{(n)} \right\rangle \boldsymbol{v}_i^{(n)}\right),\ i = 1, \ldots, k, \tag{35}$$

and we compute the orthogonal basis of the subspace $\mathcal{U}_{\mathrm{SD}}^{(n)}$ based on the normal orthogonal set $\{\boldsymbol{v}_i^{(n)}\}_{i=1}^k$. Specifically, we calculate

$$\tilde{\boldsymbol{w}}_i^{(n)} = \boldsymbol{w}_i^{(n)} - \sum_{j=1}^{k} \left\langle \boldsymbol{w}_i^{(n)}, \boldsymbol{v}_j^{(n)} \right\rangle \boldsymbol{v}_j^{(n)} - \sum_{\substack{j=1 \\ \|\tilde{\boldsymbol{w}}_j^{(n)}\| > \epsilon_w}}^{i-1} \frac{\left\langle \boldsymbol{w}_i^{(n)}, \tilde{\boldsymbol{w}}_j^{(n)} \right\rangle}{\left\| \tilde{\boldsymbol{w}}_j^{(n)} \right\|^2} \tilde{\boldsymbol{w}}_j^{(n)}, \tag{36}$$

using the method of (modified) Gram–Schmidt recursively, and define a K-column matrix ($k \leqslant K \leqslant 2k$)

$$\mathbb{U}_{\mathrm{SD}}^{(n)} = \left[\boldsymbol{v}_1^{(n)}, \ldots, \boldsymbol{v}_k^{(n)}, \tilde{\boldsymbol{w}}_i^{(n)} \Big/ \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\| : \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\| > \epsilon_w, i = 1, \ldots, k\right], \tag{37}$$

whose column vectors are orthogonal basis of the subspace $\mathcal{U}_{\mathrm{SD}}^{(n)}$ approximately. In (37), we drop $\boldsymbol{w}_i^{(n)}$ whose norm is small enough for numerical stability. We define another K-column matrix

$$\mathbb{Y}_{\mathrm{SD}}^{(n)} = \left[\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_k^{(n)}, \boldsymbol{y}_i^{(n)} : \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\| > \epsilon_w, i = 1, \ldots, k\right], \tag{38}$$

where

$$\boldsymbol{y}_i^{(n)} = \boldsymbol{H}\left(\boldsymbol{x}^{(n+1)}, \tilde{\boldsymbol{w}}_i^{(n)} \Big/ \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\|, l^{(n)}\right), \tag{39}$$

so $\mathbb{G}(\boldsymbol{x}^{(n+1)})\mathbb{U}_{\mathrm{SD}}^{(n)}$ can be approximated by $\mathbb{Y}_{\mathrm{SD}}^{(n)}$. With Rayleigh–Ritz methods, we calculate $\{\boldsymbol{\eta}_i^{(n)}\}_{i=1}^k$ as the eigenvectors corresponding to the smallest k eigenvalues of a $K \times K$ matrix $\mathbb{P}_{\mathrm{SD}}^{(n)} = (\mathbb{U}_{\mathrm{SD}}^{(n)})^\top \mathbb{Y}_{\mathrm{SD}}^{(n)}$ and accordingly set $\{\boldsymbol{v}_i^{(n+1)}\}_{i=1}^k$ as the Ritz vectors $\{\mathbb{U}_{\mathrm{SD}}^{(n)}\boldsymbol{\eta}_i^{(n)}\}_{i=1}^k$. Because of numerical errors, we use the eigenvectors of $(\mathbb{P}_{\mathrm{SD}}^{(n)} + (\mathbb{P}_{\mathrm{SD}}^{(n)})^\top)/2$ as the symmetric part of $\mathbb{P}_{\mathrm{SD}}^{(n)}$ practically.

We can implement the LOBPCG method similarly. By defining

$$\boldsymbol{w}_i^{(n)} = \boldsymbol{v}_{i-k}^{(n-1)},\ i = k+1, \ldots, 2k, \tag{40}$$

we calculate (36) and (39) for $i = 1, \ldots, 2k$ and define two K-column matrices as ($k \leqslant K \leqslant 3k$)

$$\mathbb{U}_{\mathrm{CG}}^{(n)} = \left[\boldsymbol{v}_1^{(n)}, \ldots, \boldsymbol{v}_k^{(n)}, \tilde{\boldsymbol{w}}_i^{(n)} \Big/ \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\| : \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\| > \epsilon_w, i = 1, \ldots, 2k\right], \tag{41}$$

$$\mathbb{Y}_{\mathrm{CG}}^{(n)} = \left[\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_k^{(n)}, \boldsymbol{y}_i^{(n)} : \left\| \tilde{\boldsymbol{w}}_i^{(n)} \right\| > \epsilon_w, i = 1, \ldots, 2k\right] \tag{42}$$

instead of (37) and (38). Especially, if $\|\tilde{\boldsymbol{w}}_i^{(n)}\|$ is close to ϵ_w, a reorthogonalization step for $\boldsymbol{w}_i^{(n)}$ is required for numerical stability. By Rayleigh–Ritz methods, we get $\{\boldsymbol{v}_i^{(n+1)}\}_{i=1}^k$ accordingly.

Algorithm 2. HiOSD-LOBPCG (or LOBPSD) method for a k-saddle.

Input: $k \in \mathbb{N}, l^{(0)} > 0, \boldsymbol{x}^{(0)} \in \mathcal{H}, \{\boldsymbol{v}_i^{(0)}\}_{i=1}^k \subset \mathcal{H}$ satisfying $\langle \boldsymbol{v}_j^{(0)}, \boldsymbol{v}_i^{(0)} \rangle = \delta_{ij}$.

1: Set $n = 0$, compute $\boldsymbol{f}^{(0)} = \boldsymbol{F}(\boldsymbol{x}^{(0)})$;
2: **repeat**
3: $\quad \boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \beta^{(n)} \boldsymbol{g}^{(n)}$;
4: $\quad$ Calculate $\mathbb{U}^{(n)}$ as (41) (or (37));
5: $\quad$ Calculate $\mathbb{Y}^{(n)}$ as (42) (or (38));
6: $\quad \mathbb{P}^{(n)} = \left(\mathbb{U}^{(n)}\right)^\top \mathbb{Y}^{(n)}$;
7: $\quad$ Calculate $\{\boldsymbol{\eta}_i^{(n)}\}_{i=1}^k$ as the k lowest eigenvectors of $(\mathbb{P}^{(n)} + \left(\mathbb{P}^{(n)}\right)^\top)/2$;
8: $\quad \boldsymbol{v}_i^{(n+1)} = \mathbb{U}^{(n)} \boldsymbol{\eta}_i^{(n)}$.
9: $\quad l^{(n+1)} = \max\{l^{(n)}/(1+\beta^{(n)}), \varepsilon\}$;
10: $\quad \boldsymbol{f}^{(n+1)} = \boldsymbol{F}(\boldsymbol{x}^{(n+1)})$;
11: $\quad n := n + 1$;
12: **until** $\|\boldsymbol{f}^{(n)}\| < \epsilon_F$;

Output: $\boldsymbol{x}^{(n)}, \boldsymbol{v}_1^{(n)}, \ldots, \boldsymbol{v}_k^{(n)}$.

To summarize, we present Algorithm 2 as a discrete scheme, where $\boldsymbol{g}^{(n)}$ and $\boldsymbol{d}_i^{(n)}$ are defined the same as Algorithm 1, and $\tilde{\boldsymbol{w}}_i^{(n)}$ in step 4 and $\boldsymbol{y}_i^{(n)}$ in step 5 are defined as in (36) and (39).

In each iteration step, the LOBPCG (or LOBPSD) method needs another $4k$ (or $2k$) force evaluations. Supposing $\mathbb{T}$ is the identity, the approximate eigenvectors determined in Algorithm 1 are also on the subspace $\mathcal{U}^{(n)}$ spanned by (41) (or (37)), so the LOBPCG method brings better approximate eigenvectors in each iteration step. Furthermore, the LOBPCG method ensures that $\boldsymbol{v}_i^{(n+1)}$ is a better approximation than $\boldsymbol{v}_i^{(n)}$ without a proper choice of step sizes $\gamma_i^{(n)}$. Consequently, the LOBPCG method improves accuracy and robustness at the sacrifice of computation costs.

2.4. Determination of the step size. We present several approaches to determine the step sizes in Algorithm 1 and Algorithm 2.

Explicit Euler scheme. The simplest way of choosing the step sizes is an explicit Euler scheme, $\beta^{(n)} = \gamma_i^{(n)} = \Delta t$, where Δt is a positive constant. We can accelerate the convergence by increasing Δt, but an overlarge Δt often leads to numerical instability.

Line search method. The line search method can be used to search the optimal step size in each iteration step. The step sizes of $\boldsymbol{x}$ are calculated by minimizing a value function of $\|\boldsymbol{F}(\boldsymbol{x})\|^2$, which achieves its global minima at all critical points of $E(\boldsymbol{x})$. Both exact and inexact line search methods can be applied in practice. An upper bound τ should be set for $\beta^{(n)}\|\boldsymbol{g}^{(n)}\|$ to avoid overlarge jump between states, and a lower bound should be set for $\beta^{(n)}$ so that it is possible to escape from a neighborhood of other critical points, which is different from minimization problems.

BB gradient method. By following the same spirit of the OSD [43], we adopt the BB gradient method [3] to determine the step sizes. The BB method chooses $\mathbb{H}^{(n)} = \beta^{(n)}\mathbb{I}$ as an approximation of the inverse of Hessians and imposes some quasi-Newton properties.

Let $\boldsymbol{g}^{(n)}$ denote the gradient at $\boldsymbol{x}^{(n)}$ as Algorithm 1 and 2, and define $\Delta\boldsymbol{x}^{(n)} = \boldsymbol{x}^{(n)} - \boldsymbol{x}^{(n-1)}$ and $\Delta\boldsymbol{g}^{(n)} = \boldsymbol{g}^{(n)} - \boldsymbol{g}^{(n-1)}$. By solving optimization subproblems $\min_{\beta^{(n)}} \|\Delta\boldsymbol{x}^{(n)} - \beta^{(n)}\Delta\boldsymbol{g}^{(n)}\|$ and $\min_{\beta^{(n)}} \|\Delta\boldsymbol{x}^{(n)}/\beta^{(n)} - \Delta\boldsymbol{g}^{(n)}\|$, we get the BB1 step size

$$\beta_{\mathrm{BB1}}^{(n)} = \frac{\left\langle \Delta \boldsymbol{x}^{(n)}, \Delta \boldsymbol{x}^{(n)} \right\rangle}{\left\langle \Delta \boldsymbol{x}^{(n)}, \Delta \boldsymbol{g}^{(n)} \right\rangle} \tag{43}$$

and the BB2 step size

$$\beta_{\mathrm{BB2}}^{(n)} = \frac{\left\langle \Delta \boldsymbol{x}^{(n)}, \Delta \boldsymbol{g}^{(n)} \right\rangle}{\left\langle \Delta \boldsymbol{g}^{(n)}, \Delta \boldsymbol{g}^{(n)} \right\rangle}. \tag{44}$$

In practice, since $|\langle \Delta \boldsymbol{x}^{(n)}, \Delta \boldsymbol{g}^{(n)} \rangle|$ could be small, we apply the BB2 formula (44) to solve the optimization problems (1), and the step size of $\boldsymbol{x}$ is determined as

$$\beta^{(n)} = \min \left\{ \frac{\tau}{\left\| \boldsymbol{g}^{(n)} \right\|}, \left| \frac{\left\langle \Delta \boldsymbol{x}^{(n)}, \Delta \boldsymbol{g}^{(n)} \right\rangle}{\left\langle \Delta \boldsymbol{g}^{(n)}, \Delta \boldsymbol{g}^{(n)} \right\rangle} \right| \right\} \tag{45}$$

with the same upper bound τ as the line search method, which is set as 0.5 in numerical experiments. The absolute values are taken in (45) as a precaution to avoid negative step sizes due to the negative eigenvalues of $\mathbb{G}(\boldsymbol{x})$. Similarly, the step sizes of $\boldsymbol{v}_i$ in Algorithm 1 are determined as

$$\gamma_i^{(n)} = \left| \frac{\left\langle \Delta \boldsymbol{v}_i^{(n)}, \Delta \boldsymbol{d}_i^{(n)} \right\rangle}{\left\langle \Delta \boldsymbol{d}_i^{(n)}, \Delta \boldsymbol{d}_i^{(n)} \right\rangle} \right|, \tag{46}$$

where $\Delta \boldsymbol{v}_i^{(n)} = \boldsymbol{v}_i^{(n)} - \boldsymbol{v}_i^{(n-1)}$, $\Delta \boldsymbol{d}_i^{(n)} = \boldsymbol{d}_i^{(n)} - \boldsymbol{d}_i^{(n-1)}, i = 1, 2, \ldots, k$.

3. Numerical examples. In this section, we show some numerical examples to illustrate the efficiency and reliability of the HiOSD method. The error is calculated as the gradient norm $\|\boldsymbol{F}(\boldsymbol{x})\|$ because the exact saddle points are often unavailable in practical cases. The initial directions are determined as the eigenvectors corresponding to the smallest k eigenvalues of the Hessian $\mathbb{G}(\boldsymbol{x}^{(0)})$ using the LOBPCG method and dimer approximations.

3.1. Modified Biggs EXP6 functions. We first test our algorithms on a toy model. Consider a six-dimensional Biggs EXP6 function [32],

$$B(\boldsymbol{x}) = \sum_{i=1}^{6} \left(x_3 \mathrm{e}^{-t_i x_1} - x_4 \mathrm{e}^{-t_i x_2} + x_6 \mathrm{e}^{-t_i x_5} - y_i \right)^2, \tag{47}$$

where $t_i = \frac{i}{10}$ and $y_i = \mathrm{e}^{-t_i} - 5\mathrm{e}^{-10t_i} + 3\mathrm{e}^{-4t_i}$. Because of symmetry, $B(\boldsymbol{x})$ has many global minima, such as $\hat{\boldsymbol{x}} = (1, 10, 1, 5, 4, 3)^\top$. We modify (47) with bounded quadratic arctangent terms to construct $\hat{\boldsymbol{x}}$ as a k-saddle, obtaining modified Biggs EXP6 functions

$$B_k(\boldsymbol{x}) = B(\boldsymbol{x}) - \sum_{i=1}^{k} s_i \arctan^2(x_i - \hat{x}_i) + \sum_{i=k+1}^{6} s_i \arctan^2(x_i - \hat{x}_i). \tag{48}$$

By choosing $\boldsymbol{s} = (4, 8, 16, 8, 4, 2)$, $\hat{\boldsymbol{x}}$ becomes a k-saddle of $B_k(\boldsymbol{x})$ for $k = 2, 3, 4, 5$.

For comparison, we implement the HiOSD method with the explicit Euler scheme, the HiOSD method with the BB gradient method, and the HiOSD-LOBPSD method with the BB gradient method. The HiOSD-LOBPCG method performs almost the

same as the HiOSD-LOBPSD method in this low-dimensional case because the subspaces (33) and (34) are both large enough to calculate the eigenvectors accurately. An initial point $\boldsymbol{x}^{(0)} = (0, 9, 1, 5, 4, 3)^\top$ achieves convergence to $\hat{\boldsymbol{x}}$ for all algorithms and $k = 2, 3, 4, 5$. However, $\nabla^2 B_k(\boldsymbol{x}^{(0)})$ has exactly $k-2$ negative eigenvalues, so the local information of $\boldsymbol{x}^{(0)}$ differs widely from that of $\hat{\boldsymbol{x}}$. The error tolerance ϵ_F is 10^{-10} in this example.

Because a small Δt for the explicit Euler scheme certainly leads to a large number of iteration steps, we need to determine an "optimal" step size $\widetilde{\Delta t}$, when the HiOSD method with the explicit Euler scheme converges in the fewest iteration steps. For cases of $k = 2, 3, 4, 5$, $\widetilde{\Delta t}$ is set as 0.06, 0.10, 0.12, and 0.14, respectively, and we apply these $\widetilde{\Delta t}$ to the explicit Euler scheme. For the BB gradient method, the first step size is simply set as 0.01. Results are shown in Table 1 and Figure 1. The explicit Euler scheme takes the most force evaluations even with the optimal step sizes, while the BB gradient method takes fewer iteration steps and less computation. As shown in Figure 1, the explicit Euler scheme has a linear convergent rate, while the BB gradient method has a faster linear convergent rate. With the BB gradient method, the HiOSD method costs a few more iteration steps than the HiOSD-LOBPSD method but fewer force evaluations and less CPU time. This suggests to us that though calculating unstable directions accurately can reduce the number of iterations, it may be unnecessary to calculate the unstable directions accurately when $\boldsymbol{x}$ is far from $\hat{\boldsymbol{x}}$.

To test the sensitivity and robustness of the HiOSD method, we add some perturbations on the initial point $\boldsymbol{x}^{(0)}$. When using $\boldsymbol{x}^{(0)} = (0, 9, 1, 5, 4, 3)^\top \pm 0.2 e_i$ (e_i is the ith unit coordinate vector), all HiOSD dynamics for $k = 2, 3, 4, 5$ can converge to the original saddle points. If we increase initial perturbations $\boldsymbol{x}^{(0)} = (0, 9, 1, 5, 4, 3)^\top \pm 0.5 e_i$, most dynamics still converge, but three perturbations including $+e_5, -e_6$ and $+e_6 (k = 2)$ fail to converge. If we use larger perturbations $\boldsymbol{x}^{(0)} = (0, 9, 1, 5, 4, 3)^\top \pm e_i$, more perturbations including $-e_1, -e_4, +e_5, -e_6$, and $+e_4 (k = 2), +e_6 (k = 2)$ will fail to converge.

3.2. Degenerate cases. Although the HiOSD method is developed under a minimax framework, our algorithms can be applied to some degenerate cases as well. A well-known example of degenerate saddle points is the monkey saddle [30], which is

TABLE 1

Comparison of HiOSD with Euler, HiOSD with BB, and HiOSD-LOBPSD with BB for the modified Biggs EXP6 functions B_k of $k = 2, 3, 4, 5$. Euler: the explicit Euler scheme. BB: the BB gradient method. Iter: number of iteration steps till convergence. $\boldsymbol{F}$eval: number of force evaluations till convergence.

k	Algorithm		$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2$	$\|\boldsymbol{F}(\boldsymbol{x})\|_2$	Iter	$\boldsymbol{F}$eval	Time/s
2	HiOSD	Euler	3.2e−11	9.4e−11	119	596	0.030
	HiOSD	BB	3.6e−12	2.2e−11	38	191	0.009
	HiOSD-LOBPSD	BB	1.8e−11	8.1e−11	31	260	0.011
3	HiOSD	Euler	2.3e−11	7.3e−11	71	498	0.022
	HiOSD	BB	5.5e−13	3.2e−12	36	253	0.011
	HiOSD-LOBPSD	BB	6.2e−12	5.9e−11	32	373	0.014
4	HiOSD	Euler	2.8e−11	8.9e−11	57	514	0.021
	HiOSD	BB	8.4e−12	3.1e−11	34	307	0.013
	HiOSD-LOBPSD	BB	1.9e−11	6.8e−11	31	384	0.015
5	HiOSD	Euler	1.3e−11	9.7e−11	162	1783	0.096
	HiOSD	BB	3.5e−11	8.2e−11	44	485	0.020
	HiOSD-LOBPSD	BB	6.4e−12	1.6e−11	34	429	0.018

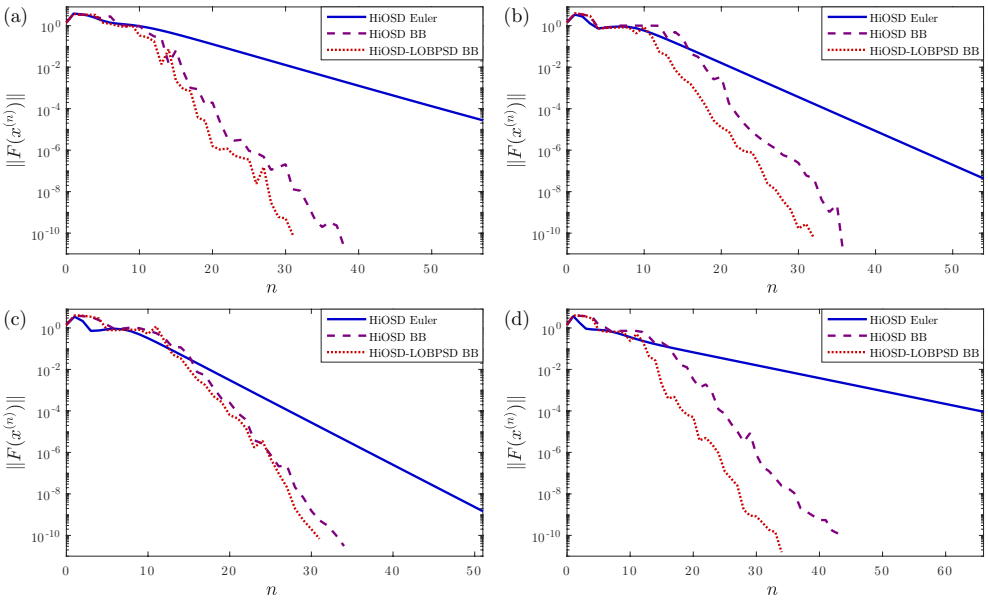

FIG. 1. *Error reduction vs. iteration steps for HiOSD with Euler, HiOSD with BB, and HiOSD-LOBPSD with BB in the modified Biggs functions* $B_k(\boldsymbol{x})$. (a) $k = 2$. (b) $k = 3$. (c) $k = 4$. (d) $k = 5$.

the origin of a two-dimensional function $E(x, y) = x^3 - 3xy^2$. It is obvious that this saddle point is not an extremum along any direction.

Now we consider computing some degenerate saddle points. The test function is a d-dimensional function ($d \geqslant 2$ is an even integer),

$$E_{d,p}(\boldsymbol{x}) = \frac{1}{p}\mathrm{Re}\sum_{j=1}^{d/2}\left(\sqrt[p]{j}\varphi(x_{2j-1}) + \frac{\mathrm{i}}{\sqrt[p]{j}}\varphi(x_{2j})\right)^p, \tag{49}$$

where $p \geqslant 3$ is an odd integer, i is the imaginary unit, and $\varphi(z) = \frac{z}{1+z^2}$. The origin is an isolated degenerate saddle point but is not a solution to any minimax problem (1). For instance, the landscape of function $E_{2,3}(\boldsymbol{x})$ is shown in Figure 2(a).

Nevertheless, we attempt to apply the HiOSD method with the explicit Euler scheme and the BB gradient method. We test our algorithms of $k = 3$ on $E_{6,3}$, and $k = 4$ on $E_{8,5}$, with an initial point $(0.4, \ldots, 0.4)^\top$. Numerical results are shown in Figure 2(b, c) with the error tolerance $\epsilon_F = 10^{-12}$ and a step size $\Delta t = 1$. Until 10000 iteration steps, the HiOSD method with the explicit Euler scheme fails to achieve the error tolerance, while that with the BB gradient method succeeds within 40 iteration steps. It shows a sublinear convergent rate of the explicit Euler scheme and a linear convergent rate of the BB gradient method. In this degenerate case, the BB gradient method shows its advantage by accelerating the convergent rate from sublinear to linear.

3.3. Lane–Emden equation. Our HiOSD method can find multiple numerical solutions of some elliptic PDEs. Consider the Lane–Emden equation [27] with zero Dirichlet boundary conditions,

$$\begin{cases} -\Delta u(\boldsymbol{x}) = u^3(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{50}$$

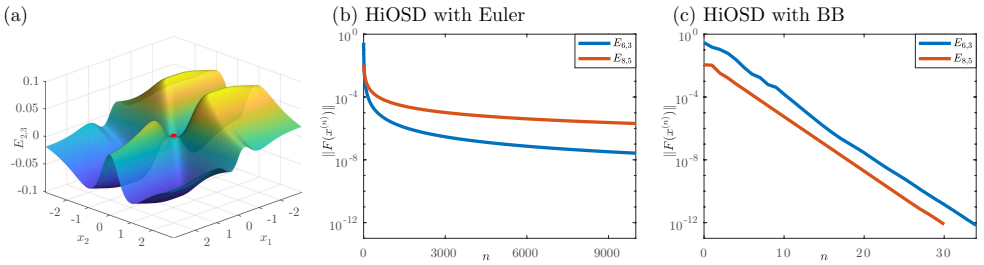

FIG. 2. (a) *The landscape of* $E_{2,3}$. *The origin (red point) is a degenerate saddle point and is not a extremum along any direction.* (b) *Error reduction vs. iteration steps of the HiOSD method with the explicit Euler scheme for degenerate functions* $E_{6,3}$ *(blue line) and* $E_{8,5}$ *(orange line).* (c) *Error reduction vs. iteration steps of the HiOSD method with the BB gradient method for the degenerate functions.*

where Ω is a bounded domain with Lipschitz boundaries. By the variational principle, solutions to the semilinear elliptic PDE (50) are critical points of the variational functional

$$E(u) = \int_\Omega \left(\frac{1}{2} |\nabla u(\boldsymbol{x})|^2 - \frac{1}{4} u^4(\boldsymbol{x}) \right) \mathrm{d}\boldsymbol{x}, \quad u \in \mathcal{H}_0^1(\Omega). \tag{51}$$

The critical points that are not local extreme points can be identified as index-k saddle points.

We use an equivalent inner product of $\mathcal{H}_0^1(\Omega)$,

$$\langle u, v \rangle = \int_\Omega \nabla u(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{52}$$

and the negative Fréchet derivative $F(u)$ with respect to the inner product (52) has the form $F(u) = -u + (-\Delta)^{-1} u^3$, where $(-\Delta)^{-1} u^3$ is the unique weak solution in $\mathcal{H}_0^1(\Omega)$ to a Poisson equation with respect to v,

$$\begin{cases} -\Delta v(\boldsymbol{x}) = u^3(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ v(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{53}$$

The solutions to (50) represent density, so only positive solutions have physical significance. Nevertheless, we regard this as an example to test our algorithms, so the solutions with negative values are acceptable as well.

The homogeneous function $u_0(\boldsymbol{x}) \equiv 0$ is a local minimizer to (50). Alternatively, several "mound" functions are defined as

$$d_i(\boldsymbol{x}) = \begin{cases} \operatorname{sign}(r_i) \left(1 + \cos \dfrac{\pi \|\boldsymbol{x} - \boldsymbol{x}_i\|_2}{\kappa r_i} \right), & \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \leqslant \kappa |r_i|, \\ 0 & \text{otherwise}, \end{cases} \tag{54}$$

where $\boldsymbol{x}_i$ and r_i are determined according to the domain Ω. We require $\{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \leqslant |r_i|\} \subset \overline{\Omega}$, and $\kappa < 1$ is set as 0.9 to ensure that boundary conditions are satisfied. Then the initial guess of a k-saddle is constructed as $\sum_{i=1}^k d_i(\boldsymbol{x}) \|\nabla d_i\|_{L^2} / \|d_i^2\|_{L^2}$.

By using the finite element method, we approximate $\Omega \subset \mathbb{R}^2$ with a polygon and partition it into n_h triangles by MATLAB subroutine `initmesh` and `refinemesh`. We use the piecewise linear function space, and the Poisson equation (53) can be numerically solved by many efficient algorithms within reasonable costs, for example, the

MATLAB subroutine `assempde`. In the following cases, all step sizes are determined by the BB gradient method with the first step size $\Delta t = 0.02$, and the error tolerance ϵ_F is set as 10^{-5}.

Case 1. Ω is an asymmetric dumbbell-shaped domain consisting of a rectangle and two circles of radii 0.5 and 1, and a mesh of 5248 triangles is generated as shown in Figure 3(a). We attempt to find four index-k saddle points with conditions

(A) $k = 2$, $\boldsymbol{x}_1 = (2, 0), r_1 = 1$, $\boldsymbol{x}_2 = (-1, 0), r_2 = 0.5$;
(B) $k = 2$, $\boldsymbol{x}_1 = (2.5, 0), r_1 = 0.5$, $\boldsymbol{x}_2 = (1.5, 0), r_2 = -0.5$;
(C) $k = 3$, $\boldsymbol{x}_1 = (2.5, 0), r_1 = 0.5$, $\boldsymbol{x}_2 = (1.5, 0), r_2 = -0.5$, $\boldsymbol{x}_3 = (-1, 0), r_3 = 0.5$;
(D) $k = 4$, $\boldsymbol{x}_1 = (2.5, 0), r_1 = 0.5$, $\boldsymbol{x}_2 = (-1.25, 0), r_3 = 0.25$,$\boldsymbol{x}_3 = (1.5, 0), r_3 = -0.5$, $\boldsymbol{x}_4 = (-0.75, 0), r_4 = -0.25$.

The numerical solutions of saddle points are shown in Figure 4, and the computation costs are shown in Table 2. The HiOSD method often takes more iteration steps but fewer force evaluations and less time because a Poisson equation is solved numerically to evaluating a force. Therefore, we apply the HiOSD method with the BB gradient method in the following cases. To test different initial guesses, we use $\sum_{i=1}^{k} 0.01 d_i(\boldsymbol{x})$ with radii $r_i = \pm 0.1$, and the center of the mound $\boldsymbol{x}_i$ is also perturbed for over

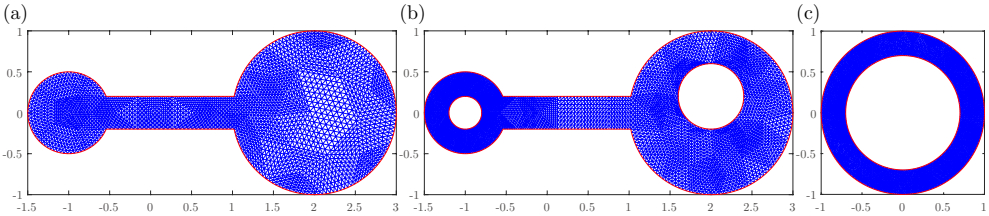

FIG. 3. *The domains of the Lane–Emden equation and their triangular partitions.* (a) *A dumbbell-shaped domain (*Case 1*).* (b) *A dumbbell-shaped domain with two cavities inside (*Case 2*).* (c) *A concentric annular domain (*Case 3*).*

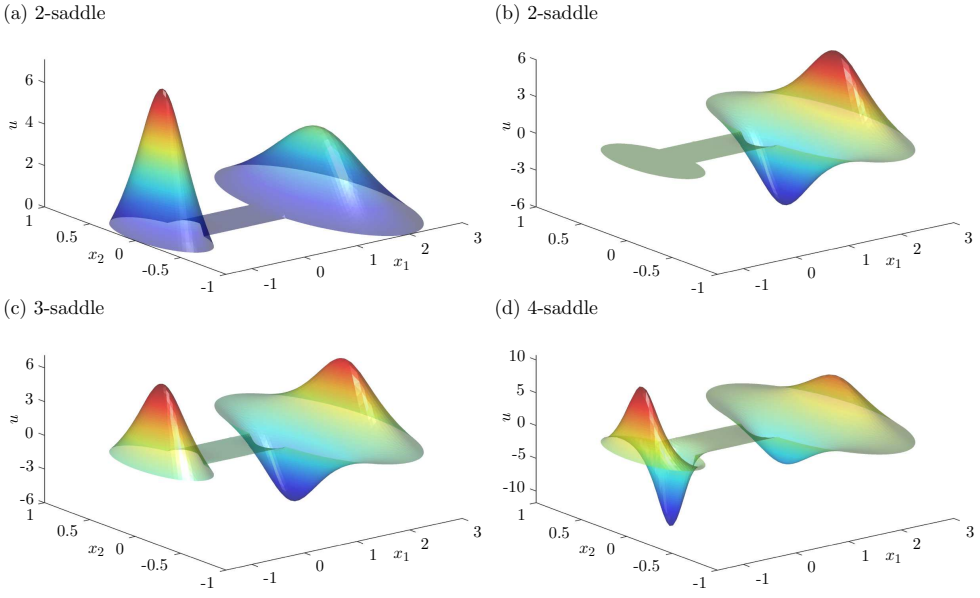

FIG. 4. *Four saddle-point solutions to the Lane–Emden equation (*Case 1*).*

TABLE 2
*Comparison of the three algorithms for the Lane–Emden equation (*Case 1*).*

	HiOSD			HiOSD-LOBPSD			HiOSD-LOBPCG		
	Iter	$\boldsymbol{F}$eval	Time/s	Iter	$\boldsymbol{F}$eval	Time/s	Iter	$\boldsymbol{F}$eval	Time/s
(A)	42	211	3.1	45	406	5.8	42	543	7.5
(B)	105	526	7.3	96	865	12.0	88	1139	15.7
(C)	102	715	10.1	86	1119	15.4	102	1933	26.7
(D)	169	1522	21.3	161	2738	37.9	157	3918	54.9

0.1 distance. The numerical tests show all perturbations can converge to the saddle points, indicating the robustness of the HiOSD method.

Case 2. Ω is the asymmetric dumbbell-shaped domain with two circular cavities of radii 0.4 and 0.2 inside, and a mesh of 9088 triangles is generated as shown in Figure 3(b). We attempt to find twelve index-k saddle points with conditions

(A) $k = 2$, $\boldsymbol{x}_1 = (2, -0.6), r_1 = 0.4$, $\boldsymbol{x}_2 = (-1.35, 0), r_2 = 0.15$;
(B) $k = 2$, $\boldsymbol{x}_1 = (2, -0.6), r_1 = 0.4$, $\boldsymbol{x}_2 = (-0.5, 0), r_2 = 0.2$;
(C) $k = 2$, $\boldsymbol{x}_1 = (-0.5, 0), r_1 = 0.2$, $\boldsymbol{x}_2 = (-1.35, 0), r_2 = 0.15$;
(D) $k = 2$, $\boldsymbol{x}_1 = (2.5, -0.4), r_1 = 0.3$, $\boldsymbol{x}_2 = (1.5, -0.4), r_2 = -0.3$;
(E) $k = 2$, $\boldsymbol{x}_1 = (-1, 0.35), r_1 = 0.15$, $\boldsymbol{x}_2 = (-1, -0.35), r_2 = -0.15$;
(F) $k = 3$, $\boldsymbol{x}_1 = (2, -0.6), r_1 = 0.4$, $\boldsymbol{x}_2 = (-1.35, 0), r_2 = 0.15$, $\boldsymbol{x}_3 = (-0.5, 0)$, $r_3 = 0.2$;
(G) $k = 3$, $\boldsymbol{x}_1 = (2.5, -0.4), r_1 = 0.3$, $\boldsymbol{x}_2 = (1.5, -0.4), r_2 = -0.3$, $\boldsymbol{x}_3 = (-1.35, 0)$, $r_3 = 0.15$;
(H) $k = 3$, $\boldsymbol{x}_1 = (2.5, -0.4), r_1 = 0.3$, $\boldsymbol{x}_2 = (1.5, -0.4), r_2 = -0.3$, $\boldsymbol{x}_3 = (-0.5, 0)$, $r_3 = 0.2$;
(I) $k = 3$, $\boldsymbol{x}_1 = (-1, 0.35), r_1 = 0.15$, $\boldsymbol{x}_2 = (-1, -0.35), r_2 = -0.15$, $x_3 = (2, -0.6), r_3 = 0.4$;
(J) $k = 4$, $\boldsymbol{x}_1 = (2.5, -0.4), r_1 = 0.3$, $\boldsymbol{x}_2 = (1.5, -0.4), r_2 = -0.3$, $\boldsymbol{x}_3 = (-1.35, 0)$, $r_3 = 0.15$, $\boldsymbol{x}_4 = (-0.5, 0), r_4 = 0.2$;
(K) $k = 4$, $\boldsymbol{x}_1 = (2.5, -0.4), r_1 = 0.3$, $\boldsymbol{x}_2 = (-1, 0.35), r_2 = 0.15$, $\boldsymbol{x}_3 = (1.5, -0.4)$, $r_3 = -0.3$, $\boldsymbol{x}_4 = (-1, -0.35), r_4 = -0.15$;
(L) $k = 4$, $\boldsymbol{x}_1 = (2.5, -0.4), r_1 = -0.3$, $\boldsymbol{x}_2 = (1.5, -0.4), r_2 = 0.3$, $\boldsymbol{x}_3 = (-1.35, 0)$, $r_3 = 0.15$, $\boldsymbol{x}_4 = (-0.5, 0), r_4 = -0.2$.

Numerical solutions of various saddle points are shown in Figure 5.

Case 3. Ω is a concentric annular domain with an inner radius 0.7 and an outer radius 1. Since Ω is geometrically symmetric, any rotation around the origin of a solution is also a solution, so a nonradial solution is degenerated. By the finite element method, a mesh of 8192 triangles is generated as shown in Figure 3(c), where this symmetry is damaged. We attempt to find four index-k saddle points with conditions

(A) $k = 2$, $\boldsymbol{x}_i = 0.85 \times (\cos\frac{2\pi i}{k}, \sin\frac{2\pi i}{k}), r_i = 0.15$;
(B) $k = 3$, $\boldsymbol{x}_i = 0.85 \times (\cos\frac{2\pi i}{k}, \sin\frac{2\pi i}{k}), r_i = 0.15$;
(C) $k = 4$, $\boldsymbol{x}_i = 0.85 \times (\cos\frac{2\pi i}{k}, \sin\frac{2\pi i}{k}), r_i = (-1)^i \times 0.15$;
(D) $k = 6$, $\boldsymbol{x}_i = 0.85 \times (\cos\frac{2\pi i}{k}, \sin\frac{2\pi i}{k}), r_i = (-1)^i \times 0.15$.

Four saddle-point solutions are shown in Figure 6.

4. Discussions and conclusions. In this paper, we propose a HiOSD method for finding index-k saddle points without evaluations of Hessians. The HiOSD method can be regarded as a generalization of the OSD method for index-1 saddle points [43]. This saddle-searching problem is formulated by a minimax optimization problem, and a HiOSD dynamical system is presented to show its linearly stable steady state is

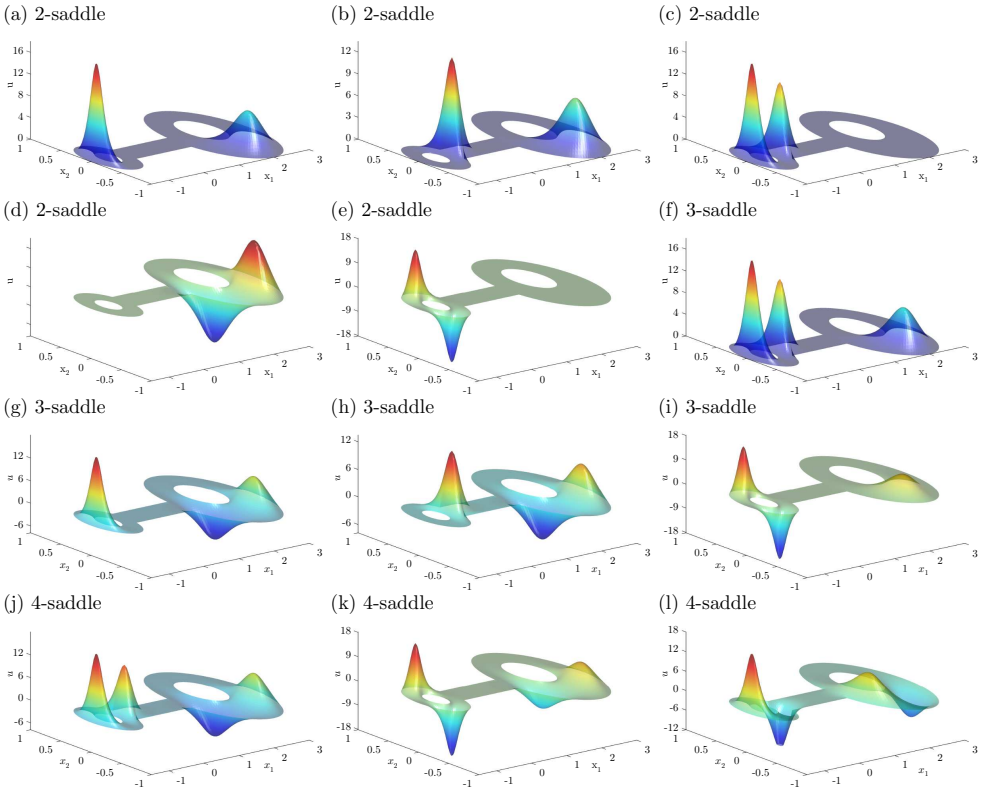

FIG. 5. *Twelve saddle-point solutions to the Lane–Emden equation (*Case 2*).*

exactly a k-saddle. We develop two algorithms as the implementation of HiOSD: one is to use simultaneous Rayleigh-quotient minimization technique, and the other is to apply the LOBPCG method in order to improve the accuracy and robustness. We also use various approaches to determine the step sizes, especially the BB gradient method. We test our algorithms by several examples to demonstrate the efficiency of the HiOSD method. By comparing the previous work by Quapp and Bofill in [33], we use dimer approximation to avoid the explicit calculation of Hessians, which is applicable to more practical problems. Under the optimization-based framework, we are able to apply efficient optimization methods to speed up the convergence and the linear stability theorem can be proved. The eigenvector calculation in the HiOSD method is a trade-off because we need to balance the computational cost and the accuracy of the eigenvector during the iterations. The adaptive strategy could be feasible to obtain the optimal convergence rate [16].

By using the HiOSD method, we provide a new approach to systematically explore the critical points of the energy functional, which can be labeled as index-k saddle points. Assuming that a k-saddle ($k \geqslant 1$) has been discovered, we can use the HiOSD algorithm to search the saddles with lower indices by following the unstable directions of this k-saddle, and repeat this searching process till reaching a minimizer. With this procedure, extensive critical points can be found systematically without initial guesses. We will pursue this idea of constructing the solution landscape that is a pathway map starting from a high-index saddle point and ending with minima elsewhere.

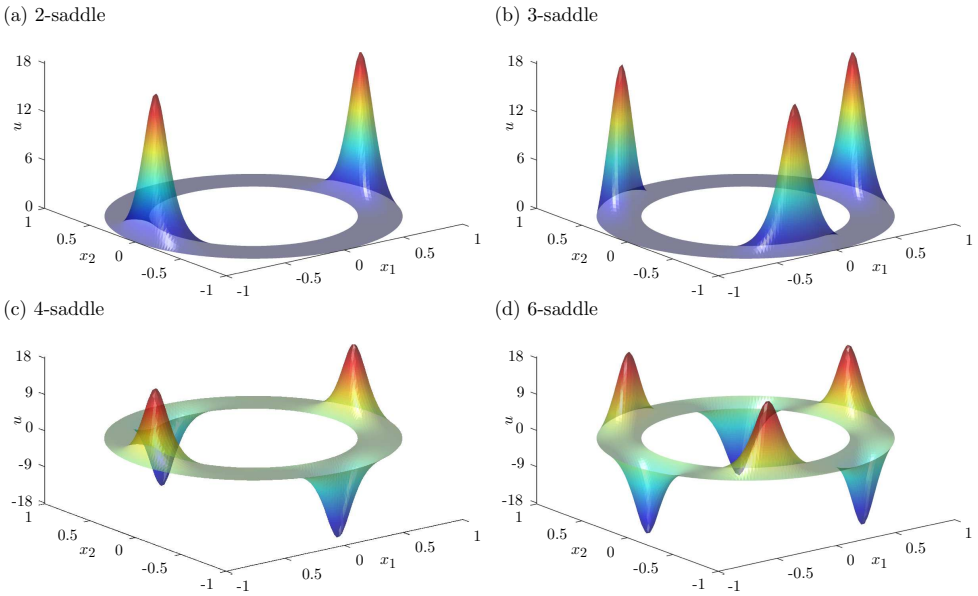

FIG. 6. *Four saddle-point solutions to the Lane–Emden equation (*Case 3*).*

The HiOSD method can be applied to solve many practical problems, for instance, finding the excited states for the Bose–Einstein condensation (BEC) [2]. A nonlinear Schrödinger equation, known as the Gross–Pitaevskii equation, can describe the dynamics of the BEC wave function. The stationary states of the BEC are the critical points of the Gross–Pitaevskii energy functional subject to a sphere constraint. Although the ground states of the BEC have been well studied [2], it is still not clear how to compute the excited states. We will develop the HiOSD method with a constraint to find the excited states of the BEC in future.

Acknowledgments. We would like to thank Professors Qiang Du, Zhijian Yang, and Weiqing Ren for fruitful discussions on the subject. We greatly appreciate Prof. Jianxin Zhou for providing their MATLAB codes of the Lane–Emden equation in [27]. We would also like to acknowledge the reviewers for their valuable comments.

REFERENCES

[1] J. BAKER, *An algorithm for the location of transition states*, J. Comput. Chem., 7 (1986), pp. 385–395.

[2] W. BAO AND Y. CAI, *Mathematical theory and numerical methods for Bose-Einstein condensation*, Kinet. Relat. Models, 6 (2013), pp. 1–135.

[3] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[4] E. CANCÈS, F. LEGOLL, M.-C. MARINICA, K. MINOUKADEH, AND F. WILLAIME, *Some improvements of the activation-relaxation technique method for finding transition pathways on potential energy surfaces*, J. Chem. Phys., 130 (2009), 114711.

[5] C. J. CERJAN AND W. H. MILLER, *On finding transition states*, J. Chem. Phys., 75 (1981), pp. 2800–2806.

[6] X. CHENG, L. LIN, W. E, P. ZHANG, AND A.-C. SHI, *Nucleation of ordered phases in block copolymers*, Phys. Rev. Lett., 104 (2010), 148301.

[7] G. M. CRIPPEN AND H. A. SCHERAGA, *Minimization of polypeptide energy:* XI. *The method of gentlest ascent*, Arch. Biochem. Biophys., 144 (1971), pp. 462–466.

[8] Q. DU AND L. ZHANG, *A constrained string method and its numerical analysis*, Commun. Math. Sci., 7 (2009), pp. 1039–1051.

[9] J. Duncan, Q. Wu, K. Promislow, and G. Henkelman, *Biased gradient squared descent saddle point finding method*, J. Chem. Phys., 140 (2014), 194102.
[10] W. E, W. Ren, and E. Vanden-Eijnden, *String method for the study of rare events*, Phys. Rev. B, 66 (2002), 052301.
[11] W. E, W. Ren, and E. Vanden-Eijnden, *Simplified and improved string method for computing the minimum energy paths in barrier-crossing events*, J. Chem. Phys., 126 (2007), 164103.
[12] W. E and E. Vanden-Eijnden, *Transition-path theory and path-finding algorithms for the study of rare events*, Annu. Rev. Phys. Chem., 61 (2010), pp. 391–420.
[13] W. E and X. Zhou, *The gentlest ascent dynamics*, Nonlinearity, 24 (2011), pp. 1831–1842.
[14] J. B. Foresman, M. Head-Gordon, J. A. Pople, and M. J. Frisch, *Toward a systematic molecular orbital theory for excited states*, J. Phys. Chem., 96 (1992), pp. 135–149.
[15] W. Gao, J. Leng, and X. Zhou, *An iterative minimization formulation for saddle point search*, SIAM J. Numer. Anal., 53 (2015), pp. 1786–1805.
[16] W. Gao, J. Leng, and X. Zhou, *Iterative minimization algorithm for efficient calculations of transition states*, J. Comput. Phys., 309 (2016), pp. 69–87.
[17] N. Gould, C. Ortner, and D. Packwood, *A dimer-type saddle search algorithm with preconditioning and linesearch*, Math. Comp., 85 (2016), pp. 2939–2966.
[18] W. J. Grantham, *Gradient transformation trajectory following algorithms for determining stationary min-max saddle points*, in Advances in Dynamic Game Theory, Birkhäuser, Boston, MA, 2007, pp. 639–657.
[19] S. Gu and X. Zhou, *Convex splitting method for the calculation of transition states of energy functional*, J. Comput. Phys., 353 (2018), pp. 417–434.
[20] D. Heidrich and W. Quapp, *Saddle points of index 2 on potential energy surfaces and their role in theoretical reactivity investigations*, Theor. Chim. Acta, 70 (1986), pp. 89–98.
[21] G. Henkelman, G. Jóhannesson, and H. Jónsson, *Methods for finding saddle points and minimum energy paths*, in Theoretical Methods in Condensed Phase Chemistry, Springer, Dordrecht, 2002, pp. 269–302.
[22] G. Henkelman and H. Jónsson, *A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives*, J. Chem. Phys., 111, pp. 7010–7022.
[23] G. Henkelman and H. Jónsson, *Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points*, J. Chem. Phys., 113 (2000), pp. 9978–9985.
[24] G. Henkelman, B. P. Uberuaga, and H. Jónsson, *A climbing image nudged elastic band method for finding saddle points and minimum energy paths*, J. Chem. Phys., 113 (2000), pp. 9901–9904.
[25] A. Heyden, A. T. Bell, and F. J. Keil, *Efficient methods for finding transition states in chemical reactions: Comparison of improved dimer method and partitioned rational function optimization method*, J. Chem. Phys., 123 (2005), 224101.
[26] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
[27] Y. Li and J. Zhou, *A minimax method for finding multiple critical points and its applications to semilinear PDEs*, SIAM J. Sci. Comput., 23 (2001), pp. 840–865.
[28] D. E. Longsine and S. F. McCormick, *Simultaneous Rayleigh-quotient minimization methods for $Ax = \lambda Bx$*, Linear Algebra Appl., 34 (1980), pp. 195–234.
[29] E. Machado-Charry, L. K. Béland, D. Caliste, L. Genovese, T. Deutsch, N. Mousseau, and P. Pochet, *Optimized energy landscape exploration using the ab initio based activation-relaxation technique*, J. Chem. Phys., 135 (2011), 034102.
[30] J. Milnor, *Morse Theory*, Princeton University Press, Princeton, NJ, 2016.
[31] R. A. Miron and K. A. Fichthorn, *The step and slide method for finding saddle points on multidimensional potential surfaces*, J. Chem. Phys., 115 (2001), pp. 8742–8747.
[32] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.
[33] W. Quapp and J. M. Bofill, *Locating saddle points of any index on potential energy surfaces by the generalized gentlest ascent dynamics*, Theor. Chem. Accounts, 133 (2014), 1510.
[34] W. Ren and E. Vanden-Eijnden, *A climbing string method for saddle point search*, J. Chem. Phys., 138 (2013), 134105.
[35] A. Samanta, M. E. Tuckerman, T.-Q. Yu, and W. E, *Microscopic mechanisms of equilibrium melting of a solid*, Science, 346 (2014), pp. 729–732.
[36] D. Sheppard, R. Terrell, and G. Henkelman, *Optimization methods for finding minimum energy paths*, J. Chem. Phys., 128 (2008), 134106.
[37] D. Wales, *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*, Cambridge University Press, Cambridge, UK, 2003.

[38] Y. Wang and J. Li, *Phase field modeling of defects and deformation*, Acta Materialia, 58 (2010), pp. 1212–1235.
[39] P. Yu, Q. Nie, C. Tang, and L. Zhang, *Nanog induced intermediate state in regulating stem cell differentiation and reprogramming*, BMC Syst. Biol., 12 (2018), 22.
[40] J. Zhang and Q. Du, *Shrinking dimer dynamics and its applications to saddle point search*, SIAM J. Numer. Anal., 50 (2012), pp. 1899–1921.
[41] L. Zhang, L.-Q. Chen, and Q. Du, *Morphology of critical nuclei in solid-state phase transformations*, Phys. Rev. Lett., 98 (2007), 265703.
[42] L. Zhang, L.-Q. Chen, and Q. Du, *Simultaneous prediction of morphologies of a critical nucleus and an equilibrium precipitate in solids*, Commun. Comput. Phys., 7 (2010), pp. 674–682.
[43] L. Zhang, Q. Du, and Z. Zheng, *Optimization-based shrinking dimer method for finding transition states*, SIAM J. Sci. Comput., 38 (2016), pp. A528–A544.
[44] L. Zhang, K. Radtke, L. Zheng, A. Q. Cai, T. F. Schilling, and Q. Nie, *Noise drives sharpening of gene expression boundaries in the zebrafish hindbrain*, Mol. Syst. Biol., 8 (2012), 613.
[45] L. Zhang, W. Ren, A. Samanta, and Q. Du, *Recent developments in computational modelling of nucleation in phase transformations*, NPJ Comput. Materials, 2 (2016), 16003.